Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2006-18

HEURISTICS FOR MULTI-PERIOD CAPACITY EXPANSION IN LOCAL TELECOMMUNICATIONS NETWORKS

by

Roxane Kouassi
Michel Gendreau
Jean-Yves Potvin
Patrick Soriano

September 2006

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone : (514) 343-7575 / Télécopieur : (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

2007-00284

PUBLICATION
CRT-2006-18

HEURISTICS FOR MULTI-PERIOD CAPACITY EXPANSION IN LOCAL TELECOMMUNICATIONS NETWORKS

by

Roxane Kouassi[1,2]
Michel Gendreau[1,2]
Jean-Yves Potvin[1,2]
Patrick Soriano[1,3]

The research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1. Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (roxane, gendreau, potvin, soriano@crt.umontreal.ca)
2. Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7
3. Service de l'enseignement des méthodes quantitatives de gestion, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2006

RÉSUMÉ

Dans cet article, différentes heuristiques sont proposées pour résoudre un problème d'expansion de la capacité dans un réseau local de télécommunications ayant une topologie arborescente. L'expansion est réalisée en installant des concentrateurs sur les sommets et des câbles sur les liens du réseau. Le but est d'identifier une stratégie d'expansion à moindre coût sur un horizon multi-périodes permettant de satisfaire la demande. Une méthode de recherche locale est d'abord développée afin d'améliorer des résultats rapportés dans la littérature sur des instances avec différentes structures de coût et de demande. Cette heuristique est ensuite intégrée au sein d'un algorithme génétique afin d'obtenir des améliorations encore plus importantes.

Mots clés : télécommunications, réseau d'accès local, expansion de la capacité, périodes multiples, recherche locale, algorithme génétique.

ABSTRACT

In this paper, different heuristics are devised to solve a multi-period capacity expansion problem for a local access telecommunications network with a tree topology. This expansion is done by installing concentrators at the nodes and cables on the links of the network. The goal is to find a least cost capacity expansion strategy over a number of periods to satisfy the demand. A local search heuristic is first proposed to improve previously reported results on problem instances based on different cost and demand structures. This heuristic is then integrated into a genetic algorithm to obtain further improvements.

Keywords: telecommunications, local access network, capacity expansion, multi-period, local search, genetic algorithm.

1 Introduction

Starting with the pioneering work in [9, 10, 19], the study of network capacity expansion problems has steadily grown over the years. Many variants of this problem can be found depending on the application considered and the type of network under study. In the telecommunications industry, a network is often composed of a backbone network, for the transfer of large volume of data, and local access networks for connecting terminals to an access node of the backbone network. In the past, researchers have mostly focused on backbone networks, while the local access networks that feed them have been relatively neglected. Also, most network planning models deal with a single time period, as opposed to more realistic multi-period problems.

In this paper, we consider the problem of finding a least-cost multi-period capacity expansion strategy for a local access network with a tree topology. Capacity expansion can be realized through the installation of flow compression devices, known as concentrators, at the nodes and cables on the links. This type of network expansion problem, like many other network design problems, is NP-hard in general.

Recent papers on capacity expansion for backbone networks may be found in [5, 6, 18], while multi-period expansion strategies are reported in [6, 11]. With regard to local access networks, most studies deal with topology design [7, 14, 15, 17]. However, in [3, 4], the authors address the capacity expansion problem via an optimization-based algorithm, where a decomposition method combines Lagrangean relaxation with a dynamic programming algorithm. The authors consider both the installation of concentrators and cables to accommodate the demand for a single time period. An exact dynamic programming algorithm for solving a similar problem is also reported in [12]. To the best of our knowledge, the work reported in [8] and [13] are the only ones that address a multi-period capacity expansion problem for local access networks. In [8], a non linear mixed-integer programming model is linearized and solved through Lagrangian relaxation. A small problem instance on a tree with 20 nodes and 4 time periods is solved with this approach. The heuristic reported in [13] can handle larger problem instances and also accounts for modular cable capacities and cost structures with economies of scale. This is an iterative procedure where, at each iteration, shadow prices are updated and used to perturb the objective in order to obtain new installation decisions. The heuristic stops when the decisions do not change after two consecutive iterations.

The work reported here addresses the problem reported in [13], except that new methodologies are proposed to improve previously reported results. The remainder of the paper is as follows. In Section 2, a mathematical programming formulation of the problem is proposed. A local search heuristic is then presented in Section 3. This is followed in Section 4 by a description of the genetic algorithm that integrates the local search heuristic. Computational experiments on problem instances based on different cost and demand structures are reported in Section 5. Finally, concluding remarks follow in Section 6.

2 Problem Formulation

We assume a local access network $G = (V, A)$ where V is the set of nodes and A is the set of arcs. The network exhibits a tree topology (by removing the orientation on the arcs) with only one node directly connected to the access node. An example is presented in Figure 1. In the figure, $Down(b) = Down(c) = a$ is the immediate downstream node of b and c, while $Down(d) = Down(e) = b$ is the immediate downstream node of d and e. Conversely, $Up(a) = \{b, c\}$ and $Up(b) = \{d, e\}$ are the sets of immediate upstream nodes of a and b, respectively.

The problem is to plan the installation of concentrators at the nodes (excluding the access node) and cables on the arcs over a number of time periods to minimize the installation costs. Furthermore, the following assumptions hold:

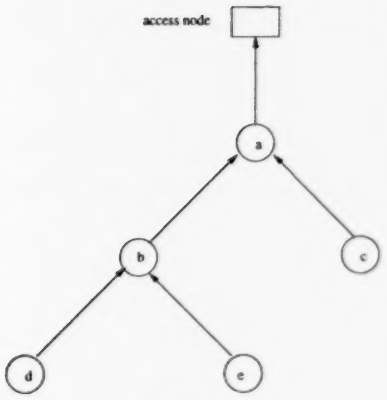

Figure 1: Local Access Network

- at each node, only one concentrator can be installed over the entire planning horizon, and this concentrator cannot be removed later on;
- on each arc, cables can only be added to existing ones over the planning horizon;
- the costs associated with concentrators and cables decrease over time, due to technology improvements; thus, it is not advantageous to install equipments too early.

In the mathematical formulation presented below, the following notation is used:

- L is the set of concentrator types, where a type is characterized by its compression factor ϕ_l, $l \in L$, $0 < \phi_l \leq 1$ (a value of 1 means that there is no flow compression),
- Φ_i is the initial concentration level at node i, $i \in V$,
- Q is the set of cable types, where a type is characterized by its bandwidth capacity β_q, $q \in Q$,
- $T = \{1, 2, ..., |T|\}$ is the set of time periods of the planning horizon,
- D_{it} is the demand at node i at period t, $i \in V$, $t \in T$,
- c_{qijt} is the cost associated with the installation of a cable of type q on arc (i, j) at period t, $q \in Q$, $(i, j) \in A$, $t \in T$,
- f_{lit} is the cost associated with the installation of a concentrator of type l at node i at period t, $l \in L$, $i \in V$, $t \in T$,

The problem can now be formulated as follows:

$$F = \min \sum_{t \in T} \sum_{l \in L} \sum_{i \in V} f_{lit} Z_{lit} + \sum_{t \in T} \sum_{q \in Q} \sum_{(i,j) \in A} c_{qijt} X_{qijt}, \tag{1}$$

subject to

$$\left(\sum_{h \in Up(i)} Y_{hit} + D_{it} \right) \left(\sum_{t' \in T, t' \leq t} \sum_{l \in L} (\phi_l - \Phi_i) Z_{lit'} + \Phi_i \right) = Y_{ijt}, \quad i \in V,\ j = Down(i),\ t \in T, \tag{2}$$

$$Y_{ijt} \leq \sum_{t' \in T, t' \leq t} \sum_{q \in Q} \beta_q X_{qijt'}, \quad (i,j) \in A, t \in T, \tag{3}$$

$$\sum_{t \in T} \sum_{l \in L} Z_{lit} \leq 1, \quad i \in V, \tag{4}$$

$$X_{qijt} \geq 0 \text{ and integer}, \quad q \in Q, (i,j) \in A, t \in T, \tag{5}$$

$$Y_{ijt} \geq 0, \quad (i,j) \in A, t \in T, \tag{6}$$

$$Z_{lit} \in \{0,1\}, \quad l \in L, i \in V, t \in T, \tag{7}$$

where

- X_{qijt} is the number of cables of type q installed on arc (i,j) at period t, $q \in Q$, $(i,j) \in A$, $t \in T$,
- Z_{lit} indicates if a concentrator of type l is installed at node i at period t, $l \in L$, $i \in V$, $t \in T$,
- Y_{ijt} is the flow on arc (i,j) at period t, $(i,j) \in A$, $t \in T$.

The flow conservation constraints (2) establish a relation between the flow that enters a node (incoming flow plus local demand) and the flow that goes out, possibly after compression. The initial concentration levels Φ_i in these equations are assumed to be 1 in the following (i.e., no flow compression in the initial state). Constraints (3) hold when the capacity installed on each arc is sufficient to handle the corresponding flow. Constraints (4) require that at most one concentrator be installed at a given node over the entire planning horizon. Note that the Z_{lit} variables can be seen as transition variables, as they indicate if a transition from the initial state takes place or not at a particular node (i.e., if a concentrator is installed at some period or not).

3 Local Search Heuristic

Starting at the root (i.e. the single node directly connected to the access node), the proposed local search heuristic explores the tree level by level. At each level, it looks for the node that provides the largest improvement when the current concentrator installation is modified with regard to the concentrator type, installation period or both. The following pseudo-code briefly sketches the general outline of the procedure:

1. Create an initial solution s which becomes the current solution;
2. While improvement do:
   - 2.1 $l \leftarrow 1$;
   - 2.2 While $l \leq l_{max}$:
     - 2.2.1 $V_{rest} \leftarrow V_l$;
     - 2.2.2 While $V_{rest} \neq \emptyset$ do:
       - a. Based on a partial evaluation, find the best move for each node $v \in V_{rest}$;
       - b. Take the $\overline{m} = \lceil |V_{rest}|/2 \rceil$ best nodes in V_{rest} according to step a. and put these nodes in $V_{\overline{m}}$;
       - c. Based on a complete evaluation, find the best move for each node $v \in V_{\overline{m}}$; put the best move in $m*$ and the associated node in $v*$.
       - d. If an improvement is obtained with move $m*$ then apply it, update current solution s, $V_{rest} \leftarrow V_{rest} - \{v*\}$; otherwise $V_{rest} \leftarrow V_{rest} - V_{\overline{m}}$;

2.2.3 $l \leftarrow l + 1$;

3. Return current solution s.

where

- l is a level in the tree (the root is assumed to be at level 1),
- l_{max} is the maximum level or depth of the tree,
- V_l is the set of nodes at level l,
- V_{rest} is the set of nodes that are still available at the current level.

Hence, the nodes at a given level are first evaluated partially to filter out unpromising ones and limit the number of complete evaluations, as explained in subsection 3.2. When the best node according to the complete evaluation is finally selected, the corresponding move is performed if it leads to an improvement. This node is then discarded and the procedure is repeated, at the same level, with the remaining nodes. If no improvement is observed, all nodes that have undergone a complete evaluation are discarded (since none of them can lead to an improvement) and the procedure is repeated, at the same level, with the remaining nodes. The current level is left only when the set of nodes at this level is exhausted. When all levels are done, the procedure is repeated anew from the root until no improvement is obtained after a complete sweep of the tree.

In the following, we describe more precisely each step of the procedure.

3.1 Initial solution

The initial solution is obtained after a single iteration of Smires' algorithm [13] (see the Appendix for a description of this algorithm). Basically, a single downstream pass is performed to take installation decisions at each node of the tree, based on null shadow prices, followed by a single upstream pass to update these prices.

3.2 Neighborhood

The neighborhood of the current solution, for a given node i, is obtained by modifying the concentrator installation with regard to the concentrator type, installation period or both. More precisely, the neighborhood of solutions is defined as follows, assuming that l_i^{old} and t_i^{old} are the current concentrator type and installation period for node i:

For l=1,...,L

For t=1,..., T

if $(l \neq l_i^{old})$ and $(t \neq t_i^{old})$ remove the current concentrator installation at node i and install a concentrator of type l at period t.

3.2.1 Partial neighbor evaluation

A partial evaluation of a neighbor solution is based only on the local impact of the new installation at node i. It involves the new concentrator cost and the cost of the cables on the outgoing arc to node $Down(i) = j$ to accommodate the new flow. Basically, the modification to the objective value, when moving from the old (current) solution to the new (neighbor) solution, is $\Delta F_i = F_i^{new} - F_i^{old}$, where

$$F_i^{new} = \sum_{t \in T} \sum_{l \in L} f_{lit} Z_{lit}^{new} + \sum_{t \in T} \sum_{q \in Q} c_{qijt} X_{qijt}^{new}$$

$$F_i^{old} = \sum_{t \in T} \sum_{l \in L} f_{lit} Z_{lit}^{old} + \sum_{t \in T} \sum_{q \in Q} c_{qijt} X_{qijt}^{old}$$

3.2.2 Complete neighbor evaluation

A complete evaluation propagates the impact of the concentrator replacement at node i, not only to the local outgoing arc, but to the whole chain of nodes and arcs leading to the root node. That is, new installation decisions are taken as we move up from node i, by applying the upstream pass procedure of Smires' algorithm [13]. Then, a downstream pass is performed to update the shadow prices on the chain. This is repeated until no more improvement is obtained. Hence, the algorithm in [13] is applied only to the chain between node i and the root node (rather than the whole tree network).

4 Genetic Algorithm

To obtain further improvements, the local search heuristic of Section 3 is used as a mutation operator within a genetic algorithm (GA). The general organization of this GA is summarized below.

1. Create an initial population of P solutions.
2. Apply the local search heuristic to each solution in the initial population.
3. The initial population becomes the current population.
4. For G iterations (generations) do
   - 4.1 Copy the P_{best}% best solutions from the current population into the new population.
   - 4.2 While the new population is not complete do:
     - 4.2.1 Select two parent solutions in the current population and apply crossover with probability p_c.
     - 4.2.2 If crossover is used, apply the local search heuristic to both offspring solutions; otherwise, copy the two parent solutions in the new population.
   - 4.3 The new population becomes the current population.
5. Return best solution found.

The main components of this GA will now be described in the following subsections.

4.1 Initial population

For creating the initial population, the algorithm reported in [13] is used. However, a single upstream and downstream pass is applied to create a solution. Different solutions are obtained by starting from different randomly generated initial shadow prices.

4.2 Selection

For selecting the parents, Stochastic Universal Sampling (SUS) is coupled with the ranking-based method [1, 2]. Basically, the solutions in the population are ranked from best (rank 1) to worst (rank P). The solution of rank 1 is assigned a MAX value and the solution of rank P is assigned a MIN value. In general, a solution of rank r is assigned the value:

$$MAX - \frac{(MAX - MIN)(r-1)}{P-1}.$$

A proportional selection scheme is then applied on these values, where the probability of selecting a solution is proportional to its value. The ranking scheme prevents the premature convergence associated with the emergence of a "super-solution" that would quickly dominate the population. Through the MIN and MAX values, it is possible to control the dispersion around the average value in the population and thus, to control the selection probabilities. In our experiments, MAX and MIN were set to 1.5 and 0.5, respectively.

4.3 Crossover

The crossover operator is based on the tree topology of the network and is somewhat inspired from genetic programming [16]. In a solution, different equipments are installed at each node and arc of the network over the time horizon (i.e., concentrators at the nodes and cables on the arcs). A node is thus randomly selected within each parent solution (the same node with regard to the network tree topology for both parents), and the equipments on the corresponding subtrees are exchanged, as illustrated in Figure 2. In this figure, a single time period is assumed and the installed equipments are indicated besides the node or arc (l identifies a concentrator and q a cable; the index is the equipment number). Thus, the equipments installed in the subtrees rooted at node b are exchanged after applying the crossover operator.

After the exchange, the two new solutions are repaired by applying the upstream pass procedure of Smires' algorithm [13]. Here, a single upstream pass is performed on the chain from the root of the subtree to the root of the network to obtain a coherent configuration of equipments. Then, a single downstream pass is performed to update the shadow prices.

4.4 Mutation

As previously mentioned, the mutation operator corresponds to the local search heuristic of Section 3. Preliminary results have led us to a variant where the local optimum obtained from a given solution does not replace the latter in the population. Only the solution value is updated (the local optimum is kept in a separate memory, though, if it corresponds to the best solution found thus far). This approach has performed better than other variants where the local optimum "takes the available slot" in the population. This is true even when the local search heuristic is applied only to a fraction of the population (to avoid premature convergence in a particular region of the search space).

4.5 Population replacement

A form of elitism is realized by copying a certain percentage of the best solutions in the current population into the new population.

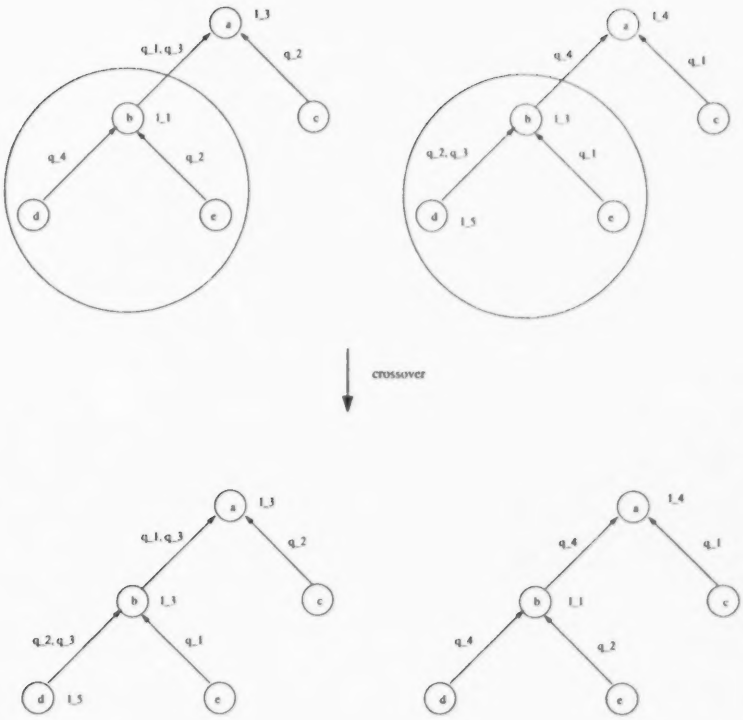

Figure 2: Crossover

5 Computational Results

In this section, computational results are reported on problems based on different cost structures and network sizes. The code was written in C++ and the tests were run on a 1.2GHz Sun workstation. We first describe the problem instances before reporting the results.

5.1 Problem instances

5.1.1 Network

We have considered three different tree structures with 110 nodes. One type is a balanced tree where all nodes, except the root and the leaves, have the same number of children; in the two other types, the number of children increases (respectively, decreases) with the node level.

5.1.2 Horizon

The planning horizon extends over $|T| = 4$ time periods.

5.1.3 Equipment

There are three different types of concentrators with compression factors of .8, .4 and .2, and four different types of cables with capacity 1, 4, 16 and 64. Their costs are modulated along three dimensions: economies of scale, obsolescence and concentrator/cable cost ratio, as it is explained below.

Economies of scale. Economies of scale are associated with increasing concentrator compression factors and cable capacities. Let us assume that cables of capacity 1, 4, 16 and 64 are associated with types q=1, 2, 3 and 4. Similarly, concentrators with compression factors 0.8, 0.4 and 0.2 are associated with types l=1, 2 and 3, respectively. At period 1, the cost of a cable of type $q+1$ is derived from the cost of a cable of type q through the following formula:

$$c_{(q+1)ij1} = \lceil r * c_{qij1} \rceil, \ (i,j) \in A,$$

where $\lceil x \rceil$ is the ceiling of x and r is a random number chosen in the interval [1,2[, [2,3[or [3,4[depending if *high*, *medium* or *low* economies of scale are considered. In the *medium* case, the cost of a cable of type $q+1$ is thus between two and three times the cost of a cable of type q, even though the capacity is increased by a factor of 4. At period 1, the formula for concentrators is:

$$f_{(l+1)i1} = \lceil r' * f_{li1} \rceil, i \in V,$$

where r' is a random multiplier in the interval [1, 1.3[, [1.3, 1.7[, or [1.7, 2.0[depending if *high*, *medium* or *low* economies of scale are considered. The costs for the other periods are derived from the obsolescence patterns described below.

Obsolescence. The cost of any given equipment type decreases from one period to the next over the planning horizon. Six different obsolescence patterns have been considered for the cables and concentrators, as shown in Table 1. The pattern 1-1-1 indicates a fixed obsolescence cost of 10% from one period to the next. The pattern 1-2-3 indicates a more significant cost decrease in the later periods, namely a reduction of 30% from period 3 to period 4, as compared to only 10% from period 1 to period 2. Conversely, the pattern 3-2-1 indicates a more significant cost decrease in the earlier periods. The patterns 2-2-2, 1-2-6 and 6-2-1 are interpreted similarly.

Pattern	Period 1→Period 2	Period 2→Period 3	Period 3→Period 4
1-1-1	10%	10%	10%
1-2-3	10%	20%	30%
1-2-6	10%	20%	60%
2-2-2	20%	20%	20%
3-2-1	30%	20%	10%
6-2-1	60%	20%	10%

Table 1: Obsolescence in percent from one period to the next

Concentrator costs versus cable costs. Three different concentrator/cable cost ratios have been considered: *high*, where concentrator costs are relatively high with regard to cable costs and lead to solutions with only a few concentrators; *low* where concentrator costs are relatively low with regard to cable costs and lead to solutions with more concentrators; and *medium* where concentrator costs are between *low* and *high*. Examples of cost structures with these three ratios for obsolescence pattern 1-2-3 and *medium* economies of scale are shown in Tables 2, 3 and 4. The cost structures associated with the other obsolescence patterns are derived similarly from the "starting" costs in period 1.

Finally, for the cables, we have set $c_{qijt} = c_{qt}, \forall (i,j) \in A$. Hence, these costs are independent of the particular arc on which the cables are installed.

cable capacity	*Period*			
	1	2	3	4
1	11	10	8	6
4	28	25	20	14
16	80	72	58	40
64	202	182	146	102
concentrator flow compression	*Period*			
	1	2	3	4
0.8	150	135	108	75
0.4	188	170	136	95
0.2	300	270	216	150

Table 2: Costs with obsolescence pattern 1-2-3, medium economies of scale and low concentrator/cable cost ratio

cable capacity	*Period*			
	1	2	3	4
1	11	10	8	6
4	28	25	20	14
16	80	72	58	40
64	202	182	146	102
concentrator flow compression	*Period*			
	1	2	3	4
0.8	200	180	144	100
0.4	260	234	188	130
0.2	422	380	304	212

Table 3: Costs with obsolescence pattern 1-2-3, medium economies of scale and medium concentrator/cable cost ratio

cable capacity	*Period*			
	1	2	3	4
1	11	10	8	6
4	28	25	20	14
16	80	72	58	40
64	202	182	146	102
concentrator flow compression	*Period*			
	1	2	3	4
0.8	300	270	216	152
0.4	386	348	278	195
0.2	562	505	404	282

Table 4: Costs with obsolescence pattern 1-2-3, medium economies of scale and high concentrator/cable cost ratio

5.1.4 Demand

The demand always increases over the planning horizon. Two different patterns have been considered: *low* demand (increase of 20% from one period to the next) and *high* demand (increase of 50% from one period to the next).

5.1.5 Parameter settings

Preliminary experiments have led to the parameter settings shown below for the GA. In particular, we were limited with regard to the population size, due to the repair mechanism after crossover and the local search-based mutation operator that proved to be quite computationally expensive. We thus used a population of 30 solutions, which allowed us to solve all problem instances in "reasonable" computation times (i.e., between half an hour and two hours). Note also that no improvement was observed beyond 10 generations.

- Population size P: 30
- Number of generations G: 10
- Crossover probability p_c: 90%
- Population replacement percentage P_{best}: 10% of best solutions in current population

5.2 Experiments

With 3 different trees, 54 cost structures (6 obsolescence patterns; 3 concentrator/cable cost ratios; 3 economies of scale) and 2 demand patterns, a total of 324 problem instances is obtained.

The results of the local search heuristic on these instances are presented in Tables 5 and 6 for low and high demand, respectively. Each number in these tables is the average of three runs, namely one run over each tree structure (comparative results for the three different tree structures are not shown, as no significant differences were observed). The first and second lines correspond to the percentage of improvement of the local search heuristic over the initial solution and over Smires' algorithm [13], respectively. The second and third line are the number of moves and computation time in seconds to reach the local optimum. Overall, the percentage of improvement of the local search heuristic over Smires' algorithm varies between 0.6% and 7.7% with an average of 3.7%. In general, larger improvements are observed when concentrator costs are high with regard to cable costs. This makes sense if we consider that the local search heuristic focuses on concentrators through its concentrator replacement move. The local search heuristic clearly dominates Smires's algorithm, as it is also less computational expensive. It requires less than 2 seconds in all cases to reach a local optimum, while the average computation time of Smires' algorithm is 5.5 seconds.

The results of the GA are presented in Tables 7 and 8 for low and high demand, respectively. The format of these tables is similar to the preceding ones, except that each entry contains the following numbers: percentage of improvement over Smires' algorithm, generation number and computation time in minutes to obtain the best solution and total computation time in minutes.

The last two tables, Tables 9 and 10, compare the percentage of improvement provided by the local search heuristic and the GA over Smires' algorithm. The GA often allows further improvements, although the magnitude of the improvement highly depends on the particular instance being solved. On some instances, the improvement obtained by the GA is more than twice that obtained by the local search heuristic. On other instances, there is no improvement. Overall, the improvement provided by the GA over the local search heuristic varies between 0% and 2.9%, with an average of 1%. It should however

be noted that the computation times of the GA are large (between half an hour and two hours) when compared with those of the local search heuristic (less than two seconds). The local search heuristic thus provides the best trade off between solution quality and computation time among the three tested heuristics.

concentrator/cable cost ratio	*economies of scale*	*obsolescence cost pattern*					
		1-1-1	1-2-3	1-2-6	2-2-2	3-2-1	6-2-1
low	low	2.3%[a]	2.2%	2.7%	2.1%	1.8%	2.2%
		1.1%[b]	1.4%	2.4%	1.7%	1.8%	1.6%
		27[c]	27	22	17	24	19
		1.6s[d]	1.7s	1.5s	1.1s	1.6s	1.3s
	medium	2.6%	3.3%	3.4%	2.9%	3.%	4.5%
		2.6%	3.3%	3.4%	2.9%	3.4%	4.5%
		13	15	13	14	15	185
		0.8s	0.8s	0.9s	0.8s	1.8s	0.9s
	high	7.3%	6.4%	5.8%	5.8%	5.3%	7.9%
		5.9%	5.2%	4.6%	4.9%	4.3%	4.0%
		13	12	11	10	10	10
		0.9s	0.9s	0.8s	0.8s	0.8s	0.6s
medium	low	4.3%	3.7%	4.2%	3.7%	3.4%	4.4%
		1.5%	1.8%	2.5%	1.8%	1.4%	2.9%
		27	28	24	25	23	21
		1.7s	1.8s	1.6s	1.7s	1.4s	1.2s
	medium	3.7%	3.0%	3.9%	3.8%	4.5%	4.1%
		3.7%	3.0%	3.9%	3.8%	4.5%	4.1%
		17	17	18	16	14	15
		1.2s	1.1s	1.3s	0.9s	0.8s	0.9s
	high	6.4%	6.9%	8.6%	8.6%	9.0%	8.6%
		4.4%	4.7%	4.4%	4.5%	3.8%	3.5%
		9	9	9	10	9	7
		0.7s	0.7s	0.7s	0.7s	0.6s	0.5s
high	low	5.7%	5.6%	6.3%	6.3%	6.2%	8.0%
		2.2%	2.0%	3.2%	3.4%	3.5%	4.2%
		23	25	25	25	24	20
		1.4s	1.5s	1.6s	1.5s	1.4s	1.3s
	medium	8.7%	8.2%	8.6%	7.8%	7.6%	5.8%
		7.6%	7.2%	7.6%	7.4%	6.7%	5.6%
		16	19	21	17	15	14
		1.0s	1.2s	1.5s	1.1s	1.0s	1.1s
	high	11.0%	9.9%	9.3%	9.5%	9.1%	8.3%
		4.2%	2.2%	2.4%	2.3%	4.7%	2.3%
		6	6	5	6	5	5
		0.5s	0.4s	0.4s	0.4s	0.4s	0.4s

[a]improvement (%) over initial solution
[b]improvement (%) over Smires' algorithm
[c]number of moves to obtain local optimum
[d]total CPU time in seconds (s)

Table 5: Local search heuristic with low demand

concentrator/cable cost ratio	*economies of scale*	*obsolescence cost pattern*					
		1-1-1	1-2-3	1-2-6	2-2-2	3-2-1	6-2-1
low	low	1.3%[a]	1.7%	1.8%	1.5%	1.4%	1.6%
		1.3%[b]	0.6%	0.7%	1.3%	1.3%	1.6%
		27[c]	23	23	19	19	24
		1.9s[d]	1.7s	1.6s	1.3s	1.3s	1.8s
	medium	3.2%	2.0%	2.3%	2.4%	2.3%	3.7%
		3.2%	2.0%	2.4%	2.4%	2.4%	3.7%
		18	17	18	15	15	13
		1.5s	1.4s	1.5s	1.2s	0.9s	0.8s
	high	7.2%	7.1%	7.7%	7.2%	7.5%	6.9%
		6.3%	6.6%	7.0%	6.5%	6.8%	4.9%
		14	12	14	15	14	13
		1.2s	1.1s	1.2s	1.3s	1.2s	1.1s
medium	low	2.6%	2.4%	2.1%	2.0%	1.6%	3.6%
		2.0%	1.2%	1.6%	1.9%	1.6%	3.6%
		26	25	25	22	16	24
		1.8s	1.8s	2.0s	1.7s	1.1s	1.6s
	medium	1.7%	2.0%	4.3%	2.4%	2.3%	3.2%
		1.7%	2.1%	4.3%	2.4%	2.3%	3.2%
		12	11	16	17	15	12
		1.0s	0.8s	1.4s	1.3s	1.1s	0.7s
	high	8.4%	8.6%	8.1%	8.5%	8.7%	6.3%
		7.7%	6.5%	6.3%	7.1%	6.5%	3.0%
		15	13	13	13	13	11
		1.3s	1.1s	1.1s	1.1s	1.2s	0.9s
high	low	3.4%	3.7%	3.7%	3.9%	3.3%	5.9%
		2.2%	2.0%	3.2%	3.4%	3.5%	4.2%
		23	25	21	25	22	23
		1.7s	1.8s	1.5s	1.9s	1.6s	1.8s
	medium	7.6%	7.2%	7.6%	7.4%	6.7%	5.6%
		5.6%	5.3%	6.6%	5.8%	4.8%	4.2%
		21	19	18	20	17	15
		1.8s	1.4s	1.4s	1.5s	1.3s	1.2s
	high	8.1%	9.5%	11.0%	11.0%	7.8%	8.3%
		4.5%	5.4%	4.4%	4.3%	4.1%	4.9%
		11	10	10	10	10	8
		1.0s	0.9s	0.9s	0.9s	0.9s	0.7s

[a] improvement (%) over initial solution
[b] improvement (%) over Smires' algorithm
[c] number of moves to obtain local optimum
[d] total CPU time in seconds (s)

Table 6: Local search heuristic with high demand

concentrator/cable cost ratio	*economies of scale*	*obsolescence cost pattern*					
		1-1-1	1-2-3	1-2-6	2-2-2	3-2-1	6-2-1
low	low	2.2%[a]	2.3%	3.2%	3.2%	4.0%	2.6%
		5[b]	3	6	4	4	3
		41m[c]	30m	48m	33m	35m	33m
		68m[d]	68m	63m	56m	60m	60m
	medium	3.8%	4.3%	4.1%	3.6%	4.1%	5.6%
		2	2	4	3	4	7
		26m	25m	41m	35m	43m	61m
		60m	60m	68m	61m	68m	71m
	high	6.0%	5.3%	4.6%	4.9%	4.3%	4.0%
		3	3	3	2	2	2
		35m	38m	33m	31m	30m	30m
		68m	73m	75m	76m	75m	75m
medium	low	3.6%	3.4%	3.7%	3.8%	3.7%	4.8%
		4	3	3	3	5	6
		33m	28m	31m	30m	35m	46m
		60m	61m	68m	55m	56m	61m
	medium	4.4%	4.7%	4.4%	4.5%	3.8%	3.5%
		2	2	2	1	1	1
		30m	31m	26m	21m	21m	15m
		68m	66m	73m	65m	68m	76m
	high	4.4%	4.7%	4.4%	4.5%	3.8%	3.5%
		2	2	2	1	1	1
		30m	31m	26m	21m	21m	15m
		68m	66m	73m	65m	68m	76m
high	low	4.5%	4.5%	5.7%	5.7%	6.1%	5.7%
		4	2	5	5	4	4
		28m	23m	35m	30m	30m	33m
		56m	56m	55m	48m	50m	65m
	medium	8.0%	7.6%	7.9%	7.6%	7.1%	5.6%
		8	6	6	7	4	1
		50m	45m	46m	53m	35m	21m
		56m	58m	63m	60m	60m	68m
	high	4.2%	2.2%	2.4%	2.3%	4.7%	2.3%
		1	1	1	1	1	1
		14m	14m	15m	15m	16m	16m
		51m	45m	86m	76m	85m	85m

[a]improvement (%) over Smires' algorithm
[b]generation number to obtain best solution
[c]CPU time to obtain best solution in minutes (m)
[d]total CPU time in minutes (m)

Table 7: Genetic algorithm with low demand

concentrator/cable cost ratio	*economies of scale*	*obsolescence cost pattern*					
		1-1-1	1-2-3	1-2-6	2-2-2	3-2-1	6-2-1
low	low	2.2%[a]	2.3%	2.1%	3.0%	3.0%	2.1%
		3[b]	5	4	3	2	6
		30m[c]	45m	45m	31m	33m	58m
		56m[d]	68m	66m	61m	63m	78m
	medium	3.7%	2.5%	3.1%	3.0%	2.9%	4.0%
		8	3	5	5	7	3
		68m	36m	56m	56m	65m	41m
		73m	71m	81m	80m	76m	86m
	high	6.8%	6.8%	7.3%	6.7%	6.9%	4.9%
		6	6	4	3	4	2
		63m	63m	56m	55m	55m	41m
		85m	83m	95m	96m	91m	106m
medium	low	3.6%	3.1%	3.1%	3.9%	3.9%	4.5%
		3	3	3	3	3	3
		36m	36m	41m	35m	36m	36m
		66m	71m	75m	65m	70m	80m
	medium	2.4%	3.3%	5.3%	3.1%	2.9%	3.5%
		6	4	4	4	7	7
		53m	41m	51m	51m	66m	73m
		70m	76m	83m	85m	80m	86m
	high	7.7%	6.5%	6.4%	7.1%	6.5%	3.0%
		3	3	3	2	3	2
		40m	43m	50m	38m	45m	33m
		85m	90m	96m	91m	95m	106m
high	low	3.9%	3.6%	3.7%	4.8%	5.6%	5.8%
		4	6	3	5	4	3
		45m	58m	43m	45m	40m	41m
		68m	76m	81m	66m	66m	78m
	medium	6.2%	5.2%	6.2%	6.0%	5.9%	4.4%
		2	5	2	3	5	6
		35m	56m	38m	36m	50m	68m
		73m	80m	83m	75m	78m	90m
	high	4.5%	5.4%	4.4%	4.5%	4.2%	5.5%
		2	2	2	3	2	1
		35m	31m	36m	36m	33m	25m
		81m	76m	86m	80m	81m	125m

[a]improvement (%) over Smires' algorithm
[b]generation number to obtain best solution
[c]CPU time to obtain best solution in minutes (m)
[d]total CPU time in minutes (m)

Table 8: Genetic algorithm with high demand

concentrator/cable cost ratio	*economies of scale*	*obsolescence cost pattern*					
		1-1-1	1-2-3	1-2-6	2-2-2	3-2-1	6-2-1
low	low	1.1%[a]	1.4%	2.4%	1.7%	1.8%	1.5%
		2.2%[b]	2.3%	3.2%	3.2%	4.0%	2.6%
	medium	2.7%	3.4%	3.5%	2.9%	3.6%	4.8%
		3.8%	4.3%	4.1%	3.6%	4.1%	5.6%
	high	5.9%	5.2%	4.6%	4.9%	4.3%	4.0%
		6.0%	5.2%	4.6%	4.9%	4.3%	4.0%
medium	low	1.5%	1.8%	2.5%	3.7%	3.7%	4.5%
		3.6%	3.4%	3.7%	3.8%	3.7%	4.8%
	medium	4.0%	3.2%	4.2%	4.0%	4.6%	4.1%
		4.4%	3.9%	4.3%	4.5%	4.7%	4.6%
	high	4.4%	4.7%	4.4%	4.5%	3.8%	3.5%
		4.4%	4.7%	4.4%	4.5%	3.8%	3.5%
high	low	2.2%	2.0%	3.2%	3.4%	3.5%	4.2%
		4.5%	4.5%	5.7%	5.7%	6.1%	5.7%
	medium	7.6%	7.2%	7.6%	7.4%	6.7%	5.6%
		8.0%	7.6%	7.9%	7.6%	7.1%	5.6%
	high	4.2%	2.2%	2.4%	2.3%	4.7%	2.3%
		4.2%	2.2%	2.4%	2.3%	4.7%	2.3%

[a]improvement (%) of local search heuristic over Smires' algorithm
[b]improvement (%) of genetic algorithm over Smires' algorithm

Table 9: Local search versus genetic algorithm for low demand

concentrator/cable cost ratio	*economies of scale*	*obsolescence cost pattern*					
		1-1-1	1-2-3	1-2-6	2-2-2	3-2-1	6-2-1
low	low	1.3%[a]	0.6%	0.7%	1.3%	1.3%	1.6%
		2.2%[b]	2.3%	2.1%	3.0%	3.0%	2.1%
	medium	3.3%	2.1%	2.4%	2.6%	2.5%	4.0%
		3.7%	2.5%	3.1%	3.0%	2.9%	4.0%
	high	6.3%	6.6%	7.0%	6.5%	6.8%	4.9%
		6.8%	6.8%	7.3%	6.7%	6.9%	4.9%
medium	low	2.0%	1.2%	1.6%	1.9%	1.7%	3.6%
		3.6%	3.1%	3.1%	3.9%	3.9%	4.5%
	medium	1.7%	2.1%	4.5%	2.5%	2.3%	3.3%
		2.4%	3.3%	5.3%	3.1%	2.9%	3.5%
	high	7.7%	6.5%	6.3%	7.1%	6.5%	3.0%
		7.7%	6.5%	6.4%	7.1%	6.5%	3.0%
high	low	1.6%	1.0%	1.2%	1.9%	3.0%	4.5%
		3.9%	3.6%	3.7%	4.8%	5.6%	5.8%
	medium	5.0%	4.2%	5.5%	5.2%	5.0%	4.2%
		6.2%	5.2%	6.2%	6.0%	5.9%	4.4%
	high	4.5%	5.4%	4.4%	4.3%	4.1%	4.9%
		4.5%	5.4%	4.4%	4.5%	4.2%	5.5%

[a]improvement (%) of local search heuristic over Smires' algorithm
[b]improvement (%) of genetic algorithm over Smires' algorithm

Table 10: Local search versus genetic algorithm for high demand

6 Conclusion

In this paper, we have presented a local search heuristic and a genetic algorithm that integrates the latter, for solving a capacity expansion problem on a local telecommunications network with a tree topology. The local search heuristic dominates the algorithm reported in [13], as it generates better solutions in less computation time. The genetic algorithm allows further improvements, but requires much larger computation times. Future work will now be aimed at refining our model to close the gap with real-world problems, in particular by allowing multiple replacements of a concentrator at a given node over the time horizon.

Acknowledgments. Financial support for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC). This support is gratefully acknowledged.

References

[1] Baker, J.E., "Adaptive Selection Methods for Genetic Algorithms", in Proceedings of the First International Conference on Genetic Algorithms and their Applications, Erlbaum, 101–111, 1985.

[2] Baker, J.E., "Reducing Bias and Inefficiency in the Selection Algorithm", in Proceedings of the Second International Conference on Genetic Algorithms and their Applications, Erlbaum, 14–21, 1987.

[3] Balakrishnan A., T.L. Magnanti, A. Shulman, R.T. Wong, "Models for Planning Capacity Expansion in Local Access Telecommunications Network", *Annals of Operations Research 33*, 239–284, 1991.

[4] Balakrishnan A., T.L. Magnanti, R.T. Wong, "A Decomposition Algorithm for Local Access Telecommunications Network Expansion Planning", *Operations Research 43*, 58–76, 1995.

[5] Chang S.-G., B. Gavish, "Telecommunications Network Topological Design and Capacity Expansion: Formulation and Algorithms", *Telecommunications Systems 1*, 99-131, 1993.

[6] Chang S.-G., B. Gavish, "Lower Bounding Procedures for Multiperiod Telecommunications Network Expansion Problems", *Operations Research 43*, 43–57, 1995.

[7] Chardaire P., "Hierarchical Two Level Location Problems", in *Telecommunications Network Planning*, B. Sanso and P. Soriano eds., Kluwer, 33–54, 1999.

[8] Chen M., A.S. Alfa, "A Model for Planning Multi-Period Local Access Communication Network", Working Paper, University of Regina, 1996.

[9] Christofides N., P. Brooker, "Optimal Expansion of an Existing Network", *Mathematical Programming 6*, 197-211, 1974.

[10] Doulliez P., R. Rao, "Optimal Network Capacity Planning: A Shortest Path Scheme", *Operations Research 23*, 811-818, 1975.

[11] Dutta A., J.-I. Lim,"A Multiperiod Capacity Planning Model for Backbone Computer Communication Networks", *Operations Research 40*, 689–705, 1992.

[12] Flippo O.E., A.W.J. Kolen, A.M.C.A. Koster, R.L.M.J. van de Leensel, "A Dynamic Programming Algorithm for the Local Access Telecommunication Network Expansion Problem", *European Journal of Operational Research 127*, 189–202, 2000.

[13] Gendreau M., Potvin J.-Y., Smires A., Soriano P., "Multi-Period Capacity Expansion for a Local Access Telecommunications Network", forthcoming in *European Journal of Operational Research*, 2006.

[14] Gouveia L., J. Paixão, "Dynamic Programming Based Heuristics for the Topological Design of Local Access Networks", *Annals of Operations Research 33*, 305–327, 1991.

[15] Gouveia L., M.J. Lopes, "Using Generalized Capacitated Trees for Designing the Topology of Local Access Networks", *Telecommunications Systems 7*, 315–337, 1997.

[16] Koza J., Genetic Progamming, The MIT Press, 1992.

[17] Mateus G.R., R.V.L. Franqueira, "Model and Heuristic for a Generalized Access Network Design Problem", *Telecommunications System 15*, 257–271, 2000.

[18] Parrish S, T. Cox, W. Kuehner, Y. Qiu, "Planning for Optimal Expansion of Leased Line Communications Networks" *Annals of Operations Research 36*, 347–364, 1992.

[19] Zadeh N., "On Building Minimum Cost Communication Networks over Time", *Networks 4*, 19-34, 1974.

Appendix

The algorithm reported in [13] is an iterative heuristic where, at each iteration, new installation decisions are taken at each node of the tree based on prices that are modified from one iteration to the next. The algorithm stops when an equilibrium point is attained (i.e., when the prices do not change anymore), or when a given number of iterations is reached. At each iteration, a downstream pass for generating installation decisions is followed by an upstream pass for adjusting the prices, as described below.

A.1 Downstream pass

At each iteration k, local installation decisions are taken at every node of the network. This is done by starting at the leaves and proceeding downstream until the node directly connected to the access node is reached. These local decisions relate to the (possible) installation of a concentrator at a node and cables on the outgoing arc to accommodate the flow. As it is necessary to know the amount of incoming flow to take appropriate installation decisions, a node can be processed only when all upstream nodes have already been processed.

Assuming that the incoming flow at node i at each time period is known (i.e., installation decisions have already been taken for all upstream nodes), decisions about the installation of a concentrator at the node and cables on the outgoing arc, over the planning horizon, are taken to minimize:

$$F_i^k = \sum_{t\in T}\sum_{l\in L} f_{lit} Z_{lit} + \sum_{t\in T}\sum_{q\in Q} c_{qijt} X_{qijt} + \sum_{t\in T} p_{jt}^k Y_{ijt},\ i \in V,$$

where j=$Down(i)$ is the immediate downstream node of i and p_{jt}^k is the current price (at iteration k) charged by node j to node i for each unit of flow sent by i at time period t. The first component of F_i^k is the installation cost of a concentrator at a given time period, if any (at most one Z_{lit} variable can be set to 1 for each node i); the second component corresponds to the installation cost of all cables that are necessary to accommodate the flow Y_{ijt} on arc (i,j), $t \in T$; the third component is the total price charged by node j for the flow Y_{ijt} sent on arc (i,j), $t \in T$. Note that the prices p_{jt}^k are initially set to $p_{jt}^0 = 0, j \in V, t \in T$, but are modified from one iteration to the next to induce different installation decisions.

We find the minimum of F_i^k by considering, in turn, the installation of each type of concentrator at each period of the planning horizon and we return the best alternative at the end.

A.2 Upstream pass

When the downstream pass at iteration k is completed and installation decisions have been taken at every node of the network, an upstream pass takes place to modify the current prices and get the prices $p_{jt}^{k+1}, j \in V, t \in T$ to be used at the next iteration. This upstream pass starts at the node directly connected to the access node and ends at the leaves. Basically, every node j charges to its immediate upstream nodes $i \in Up(j)$ a price p_{jt} for each unit of flow sent on link (i,j) at period t. The prices per unit of flow at iteration $k+1$ are determined as follows:

$$p_{jt}^{k+1} = \frac{\sum_{l\in L} f_{ljt} Z_{ljt}^{k\star} + \sum_{q\in Q} c_{qjrt} X_{qjrt}^{k\star} + p_{rt}^{k+1} Y_{jrt}^{k\star}}{\sum_{i\in Up(j)} Y_{ijt}^{k\star} + D_{jt}},\ r = Down(j),\ j \in V,\ t \in T.$$

In this equation, the numerator is the cost associated with node j at period t, where $Z^{k\star}$, $X^{k\star}$ and $Y^{k\star}$ denote the variable values in the solution obtained at the end of the downstream pass. Clearly,

as the updated prices per unit of flow at the immediate downstream node r are needed to update the prices at node j, the latter can only be processed when all its downstream nodes have been processed. The denominator is the total flow handled by node j, that is the incoming flow plus the local demand. Thus, node j charges to each immediate upstream node the cost it incurs to process each unit of flow that goes through it. One exception is the access node, which does not charge anything to its unique immediate upstream node.

An example, based on a single time period $t = 1$, is provided in Figure 3 at some iteration k. Here, the downstream pass has just been completed and we know what the network installations are, as well as the flow on each arc (c.f., labels associated with full arrows). Now, we consider node a, which is directly connected to the access node. The flow going through this node is equal to the incoming flow from nodes b and c, $10 + 20 = 30$, plus a local demand of 20, for a total of 50. As the outgoing flow is only 20, there is clearly a concentrator at this node. Each dashed arrow between two nodes y and x is labeled with the price charged by y to x, namely the price per unit of flow times the flow on arc (x, y). If we assume that the equipment costs (concentrator, cables) installed at node a is 100, then $p_{a1}^{k+1} = \frac{100}{10+20+20} = 2$ per unit of flow. Node a will thus charge a price of $10 * 2 = 20$ to node b and $20 * 2 = 40$ to node c at iteration $k + 1$. A price of $20 * 2 = 40$ is also associated with the local demand at node a. If we sum up these three values, we obtain $20 + 40 + 40 = 100$, the total cost incurred by node a to process its flow. When this calculation is done, we can proceed with nodes $b, c \in Up(a)$. If we assume that the cost of the equipments installed at node b is 200, then $p_{b1}^{k+1} = \frac{200+20}{10+20+10} = 5.5$ per unit of flow (note that the price charged by node a to node b is added to the equipment costs in the numerator). Node b will thus charge a price of $10 * 5.5 = 55$ to nodes $d, e \in Up(b)$ at iteration $k + 1$. A price of $20 * 5.5 = 110$ is also associated with the local demand at node b. Once again, we have $55 + 55 + 110 = 200 + 20 = 220$. This procedure is repeated until all nodes have been processed.

A.3 Iterative heuristic

The complete heuristic is now described, where k_{max} is a fixed number of iterations:

1. $k \leftarrow 0$.
2. *changeflag*$\leftarrow 1$.
3. $p_{jt}^0 \leftarrow 0,\ j \in V,\ t \in T$.
4. While $(k \leq k_{max})$ and $(changeflag = 1)$ do
   - 4.1 Downstream pass to produce a solution using $p_{jt}^k, j \in V,\ t \in T$.
   - 4.2 Upstream pass to calculate $p_{jt}^{k+1},\ j \in V,\ t \in T$.
   - 4.3 If $p_{jt}^k = p_{jt}^{k+1},\ \forall j \in V,\ \forall t \in T$ then *changeflag*$\leftarrow 0$.
   - 4.4 $k \leftarrow k + 1$.
5. Return best solution found.

At each iteration, the heuristic restarts "from scratch" and takes new installation decisions at each node, but with different prices to guide the decision process. Note that once identical prices (or, equivalently, identical solutions) are obtained for two consecutive iterations, no further price changes are possible.

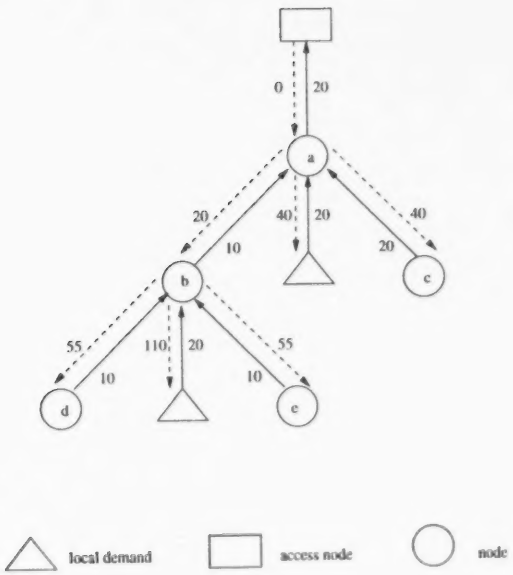

Figure 3: Flows and prices